Open access • Proceedings Article • DOI:10.1109/CVPR.2013.62

Scalable Sparse Subspace Clustering — Source link

Xi Peng, Lei Zhang, Zhang Yi

Institutions: Sichuan University

Published on: 23 Jun 2013 - Computer Vision and Pattern Recognition

Topics: Cluster analysis, Correlation clustering, Fuzzy clustering, Spectral clustering and Graph (abstract data type)

Scalable Sparse Subspace Clustering

Xi Peng, Lei Zhang and Zhang Yi
Machine Intelligence Laboratory, College of Computer Science, Sichuan University,
Chengdu, 610065, China.
pangsaai@gmail.com, {leizhang, zhangyi}@scu.edu.cn

Abstract

In this paper, we address two problems in Sparse Subspace Clustering algorithm (SSC), i.e., scalability issue and out-of-sample problem. SSC constructs a sparse similarity graph for spectral clustering by using ℓ^1-minimization based coefficients, has achieved state-of-the-art results for image clustering and motion segmentation. However, the time complexity of SSC is proportion to the cubic of problem size such that it is inefficient to apply SSC into large scale setting. Moreover, SSC does not handle with out-of-sample data that are not used to construct the similarity graph. For each new datum, SSC needs recalculating the cluster membership of the whole data set, which makes SSC is not competitive in fast online clustering. To address the problems, this paper proposes out-of-sample extension of SSC, named as Scalable Sparse Subspace Clustering (SSSC), which makes SSC feasible to cluster large scale data sets. The solution of SSSC adopts a "sampling, clustering, coding, and classifying" strategy. Extensive experimental results on several popular data sets demonstrate the effectiveness and efficiency of our method comparing with the state-of-the-art algorithms.

1. Introduction

Clustering is one of the fundamental and important topics in pattern recognition and computer version communities, which aims to group the similar patterns into the same cluster by maximizing the inter-cluster dissimilarity and the intra-cluster similarity. Over the past two decades, a number of clustering approaches such as k-means clustering have been extensively studied. Recently, clustering in non-linearly separable data has became a hot topic. To address this problem, numerous methods have been proposed, for example, kernel-based clustering [10], algebraic methods [20], iterative methods [27], statistical methods [19], and spectral clustering [21]. In this paper, we mainly focus on scalable spectral clustering algorithms.

Spectral clustering belongs to the family of subspace clustering [26] which aims at finding a low-dimensional subspace for each group of points. The assumption of spectral clustering is that the high-dimensional data actually lie on a low-dimensional manifold. It has achieved impressive results in various applications [2, 5]. The basic idea of spectral clustering is to find a cluster membership of the data points by using the spectrum of the affinity matrix that depicts the similarity among data points, and thus the construction of similarity graph lies on its heart. In a similarity graph, the vertex denotes a data point and the connection weight between two points represents the similarity.

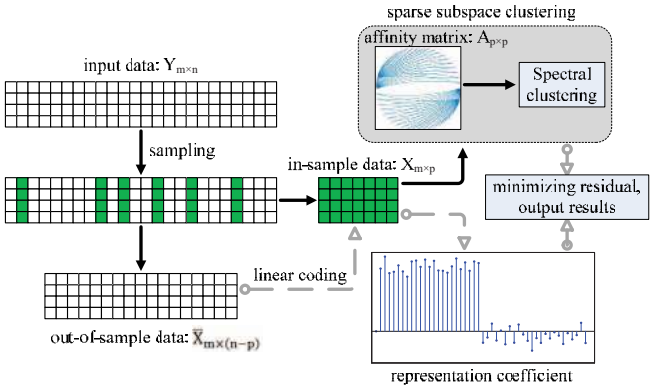


Figure 1. Architecture of our Scalable Sparse Subspace Clustering algorithm.

Generally, there are two ways to build a similarity graph. One is based on pairwise distance, e.g. Euclidean distance. However, the points are close in terms of pairwise distance may not belong to the same subspace. As a result, the way to construct similarity graph by using reconstruction representation coefficient has became more and more popular since it measures the similarity based on the data distribution. The representation coefficient based spectral clustering approaches assume that each data point can be denoted as a linear combination of other data points owing to the intrinsic similarity among the intra-subspace data, and thus the representation coefficient can be regarded as a kind of measurement. Several algorithms have been proposed, for

example, Locally linear manifold clustering (LLMC) [13], SMCE [8], ℓ^1-graph [5, 30], Low Rank Representation (LRR) [18] and L2-graph [24].

Recently, Elhamifar and Vidal [7, 9] constructed a sparse similarity graph by using ℓ^1-minimization based coefficients for spectral clustering, named Sparse Subspace Clustering (SSC). It automatically selects the nearby points for each datum by utilizing the principle of sparsity without a fixed global parameter to determine the size of neighborhood. However, SSC requires solving n optimization problems over n data points and calculating the eigenvectors of the graph Laplacian matrix. Its computational complexity is more than $O(n^3)$ even though the fast ℓ^1-solver is used, which means that any medium sized data set will bring up the scalability issues with SSC. Moreover, SSC is an offline algorithm which does not handle with the data not used to construct similarity graph (out-of-sample data). For each new datum, SSC has to perform the algorithm over the whole data set such that it is not suitable for fast online clustering.

In this paper, we propose a simple but effective out-of-sample extension of SSC, named as Scalable Sparse Subspace Clustering (SSSC), which resolves the scalability issue in SSC as a kind of out-of-sample problem. The proposed method adopts a "sampling, clustering, coding, and classifying" strategy, as shown in Figure 1. Our motivation derives from the sparsity assumption that each data point can be represented as a linear combination of a few basis vectors. It implies that the union of the linear subspaces spanned by in-sample data could equal or approximate to that spanned by the original data. In other words, one could use a small number of data points (in-sample data) to represent the original one without loss of information. Consequently, in theory, the solution of scalability issue may be not at the cost of clustering quality, which is an interesting conclusion. The property may only belong to the representation coefficient based spectral clustering, which has not been exploited to develop a scalable method as far as we known.

The rest of the paper is organized as follows: Section 2 provides a brief review of SSC and some popular methods for large scale spectral clustering. Section 3 presents the Scalable Sparse Subspace Clustering (SSSC) method. Section 4 carries out the experiments to examine the effectiveness of SSSC. Finally, Section 5 concludes this work.

2. Related Works

Except in some specified cases, **lower-case bold letters** represent column vectors and **upper-case bold ones** represent matrices. $\mathbf{A}^T$ denotes the transpose of the matrix $\mathbf{A}$ whose pseudo-inverse is $\mathbf{A}^{-1}$, and $\mathbf{I}$ is reserved for identity matrix. Moreover, we summarize some notations used throughout the paper in Table 1.

Table 1. Notations used in this paper.

Notation	Definition
n	the number of data points
m	the dimensionality of data
k	the number of clusters
p	the number of in-sample data
$c_{i,j}$	the jth element of $\mathbf{c}_i$
$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$	data points
$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]$	the sparse representation of $\mathbf{Y}$
$\mathbf{Y}_i = \mathbf{Y} \setminus \mathbf{y}_i$	the data points except $\mathbf{y}_i$
$\mathbf{A} \in \mathbb{R}^{p \times p}$	affinity matrix
$\mathbf{L} \in \mathbb{R}^{p \times p}$	Laplacian matrix
$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$	the first k eigenvectors of $\mathbf{L}$
$\mathbf{V} \in \mathbb{R}^{p \times k}$	eigenvector matrix
$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p]$	in-sample data
$\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots, \bar{\mathbf{x}}_{n-p}]$	out-of-sample data

2.1. Sparse Subspace Clustering

Recently, some researchers have explored to utilize the inherent sparsity of sparse representation to construct a similarity graph for dimension reduction [25], image analysis [5], and so on. In these works, Elhamifar and Vidal [7, 9] proposed the SSC algorithm for subspace segmentation with well-founded recovery theory for independent subspaces and disjoint subspaces. SSC calculates the similarity among data points via solving the following optimization problem:

$$\min \|\mathbf{c}_i\|_1 \qquad \text{s.t. } \|\mathbf{y}_i - \mathbf{Y}_i\mathbf{c}_i\|_2 < \delta, \tag{1}$$

where $\mathbf{c}_i \in \mathbb{R}^n$ is the sparse representation of data point $\mathbf{y}_i \in \mathbb{R}^m$ over dictionary $\mathbf{Y}_i \triangleq [\mathbf{y}_1 \ldots \mathbf{y}_{i-1}\ \mathbf{0}\ \mathbf{y}_{i+1} \ldots \mathbf{y}_n]$, and $\delta \geq 0$ is the error tolerance. The solution of the (1) can be achieved by using convex optimization methods referring to [31] which provides an extensive survey.

After getting the coefficients of all data points, SSC performs spectral clustering [21] over the sparse coefficients as described in Algorithm 1.

It is easy to find that the computational complexity of SSC is very high. For example, SSC needs $O(tn^2m^2 + tmn^3)$ to construct a similarity graph even though it adopts Homotopy optimizer [23] to get the sparsest solution, where Homotopy optimizer is one of the fastest ℓ^1-minimization algorithm according to [31] and t denotes the number of iterations of the algorithm. In addition, it will take $O(n^3)$ to calculate the eigenvectors of the Laplacian matrix $\mathbf{L}$. Consider $\mathbf{L}$ is a sparse matrix, the time complexity of this step could be reduced to $O(mn + mn^2)$ when Lanczos eigensolver is used. However, it is still a daunting task even for a moderate $n > 100,000$.

Algorithm 1 Sparse Subspace Clustering (SSC).

Input: A set of data points $\mathbf{Y} \in \mathbb{R}^{m \times n}$, and the number of desired clusters k.

1: Solve the ℓ^1-minimization problem (1) to get the collection of $\mathbf{c}_i$.
2: Form an affinity matrix $\mathbf{A} = |\mathbf{C}|^T + |\mathbf{C}|$, where $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]$.
3: Construct a Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ using $\mathbf{A}$, where $\mathbf{D} = \mathrm{diag}\{d_i\}$ with $d_i = \sum_{j=1}^{n} \mathbf{A}_{ij}$.
4: Obtain the eigenvector matrix $\mathbf{V} \in \mathbb{R}^{n \times k}$ which consists of the first k normalized eigenvectors of $\mathbf{L}$ corresponding to its k smallest eigenvalues.
5: Get the segmentations of the data by performing k-means on the row of $\mathbf{V}$.

Output: The cluster assignments of $\mathbf{Y}$.

2.2. Large Scale Spectral Clustering

Recently, some works have devoted to solve the scalability issue in spectral clustering. One natural option is to reduce the time cost of eigen-decomposition over Laplacian matrix. Fowlkes et al. [11] proposed using Nyström method to avoid computing the whole similarity matrix. Chen et al. [3] performed eigen-decomposition in a distributed systems.

Another option is to reduce the data size by performing sampling techniques or replacing the original data set with a small number of points. Yan et al. [30] provided a framework for fast approximate spectral clustering by selecting some representative points based on k-means or random projection trees. Chen and Cai [4] proposed an approach, called Landmark-based Spectral Clustering (LSC). It firstly chooses p representative points (landmarks) from data using k-means clustering or randomly sampling; then, constructs a Laplacian matrix $\mathbf{L} = \mathbf{A}^T\mathbf{A}$, where the element $\mathbf{a}_{ij}$ of $\mathbf{A} \in \mathbb{R}^{p \times n}$ is the similarity between the original data point and the landmark based on pairwise distance. finally, performs spectral clustering over $\mathbf{L}$. Wang et al. [28] firstly selected landmarks by performing selective sampling technique; then performed spectral clustering over the chosen samples based on pairwise distance; after that, projected out-of-sample data into a low-dimensional space using Locality Preserving Projections algorithm [16]; finally, got the labels of out-of-sample data by using k-nearest neighbor classifier in the embedding space. Based on similar idea, Nie et al. [22] proposed Spectral Embedded Clustering (SEC) which groups out-of-sample data by using subspace learning methods.

The second option, which selects some key data points to represent the others, has became more and more popular owing to its effectiveness and efficiency. However, all these approaches focus on developing a large scale method for pairwise similarity based spectral clustering and did not explore the intrinsic characteristics of data structure. In this paper, we will fill this gap by making SSC feasible for grouping out-of-sample data. As far as we know, it is the first work to address the scalability issue of non-pairwise similarity based spectral clustering.

3. Scalable Sparse Subspace Clustering

In this section, we present the Scalable Sparse Subspace Clustering algorithm (SSSC) which is an out-of-sample extension of SSC [7, 9]. We make SSC feasible to cluster large scale data sets in "sampling, clustering, coding, and classifying" manner. The first two steps chose a small number of data points as in-sample data and perform SSC over it. The third and fourth steps encode out-of-sample data as a linear combination of in-sample data and assign the non-sampled data to the cluster which produces the minimal residual over the chosen data, respectively.

3.1. Algorithm Description and Discussion

The basic idea of our approach is: for a set of data points $\mathbf{Y}$ drawn from the linear subspaces $\{S_i\}_{i=1}^{k}$, each subspace S_i is spanned by a collection of data points $\mathbf{B}_i \triangleq \{\mathbf{y}_i\}_{i=1}^{d_i}$, where d_i is the dimensionality of S_i. Then, any data points $\mathbf{y}_i \in S_i$ and $\mathbf{y}_i \notin \mathbf{B}_i$ can be represented as a linear combination of $\mathbf{B}_i$. It implies that scalable spectral clustering could be resolved in two steps.

The first step is performing spectral clustering over a small number of data points (in-sample data) which could exactly or approximately represent the original data space. In other words, out-of-sample data has no influence on the segmentation result. The assumption is reasonable and has been widely used in many works, e.g., Neighbor Preserving Embedding [15]. It is an interesting and challenging problem to select some key points as in-sample data. Benefit from the characteristic of large scale data set, we adopt uniform random sampling technique which has been proved competitive in practice [4, 6, 30].

After getting the cluster assignment of in-sample data, it is nature to obtain the cluster membership of out-of-sample data by performing classification over it. Based on the above analysis, we can find that the clustering error of SSSC mainly comes from the grouping error of out-of-sample data. Therefore, it is a key to design an effective approach for grouping the non-sampled data. The most simple method is to directly assign out-of-sample data to the nearest cluster in terms of Euclidean distance or other pairwise distances. However, most high-dimensional data are not lie into the Euclidean space such that Euclidean distance is not a good metric to measure the adjacency relationship among data points. On the other hand, a important task of subspace clustering is to find a low-dimensional representation for each data point. Therefore. we adopted sparse representa-

tion based classification (SRC) [29] which could satisfy the requirements. The first step of SRC is coding the testing sample $\mathbf{y}$ over the training data $\mathbf{D}$:

$$\min \|\mathbf{c}\|_1 \qquad \text{s.t. } \|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2 < \delta, \tag{2}$$

where the columns of $\mathbf{D}$ are sorted according to their labels.

Once the optimal $\mathbf{c}$ is achieved, SRC assigns $\mathbf{y}$ to the class that has the minimum residual:

$$r_i(\mathbf{y}) = \|\mathbf{y} - \mathbf{D} \cdot \delta_i(\mathbf{c}^*)\|_2. \tag{3}$$

$$\text{identity}(\mathbf{y}) = \underset{i}{\text{argmin}}\{r_i(\mathbf{y})\}, \tag{4}$$

where the nonzero entries of $\delta_i(\mathbf{c}^*) \in \mathbb{R}^n$ are the elements in $\mathbf{c}^*$ that are associated with ith class, and the $\text{identity}(\mathbf{y})$ denotes the label of $\mathbf{y}$.

Although SRC has achieved impressive results in pattern recognition, a recent work [32] empirically showed that non-sparse linear representation could achieve competitive recognition rate with less time cost. Therefore, to reduce the computational cost for grouping out-of-sample data, the main block in large scale clustering, we perform linear coding scheme but sparse one by solving

$$\min_{\mathbf{c}} \|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2^2 + \lambda\|\mathbf{c}\|_2^2, \tag{5}$$

where the second term is used to avoid over-fitting. The solution of (5) is named as Collaborative Representation by Zhang et al. [32].

Then, the classification results are achieved by calculating regularized residuals over all classes by computing

$$r_i(\mathbf{y}) = \frac{\|\mathbf{y} - \mathbf{D}\delta_i(\mathbf{c}^*)\|_2}{\|\delta_i(\mathbf{c}^*)\|_2}. \tag{6}$$

and assigning $\mathbf{y}$ to the class which produces the minimal $r_i(\mathbf{y})$ using (4).

Algorithm 2 summarizes our algorithm.

3.2. Complexity Analysis

Suppose p samples are randomly selected from n data points with dimensionality m, we need $O(t_1p^2m^2 + t_1mp^3 + p^2 + t_2pk^2)$ to get the cluster membership over in-sample data when Homotopy optimizer [23] is used to solve ℓ^1-minimization problem and Lanczos eigensolver is used to compute the eigenvectors of Laplacian matrix $\mathbf{L}$, where k is the number of desired clusters, and t_1 and t_2 are the number of iterations of Homotopy optimizer and k-means clustering, respectively.

To group out-of-sample data points, our approach needs computing the inverse of the matrix $\mathbf{X}\mathbf{X}^T$ to get the linear representation of $\bar{\mathbf{X}} \in \mathbb{R}^{m\times(n-p)}$. Therefore, the time complexity is $O(pm^2 + p^3 + (n-p)p^2)$.

Algorithm 2 Scalable Sparse Subspace Clustering (SSSC).

Input: A set of data points $\mathbf{Y} \in \mathbb{R}^{m\times n}$, the number of desired clusters k, and the rigid regression parameter $\lambda = 1e-6$.

1: Select p data points from $\mathbf{Y}$ using random sampling or other methods, e.g. k-means clustering, denoted by $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p)$.

2: Perform SSC (Algorithm 1) over $\mathbf{X}$.

3: Calculate the linear representation of out-of-sample data $\bar{\mathbf{X}}$ over $\mathbf{X}$ by

$$\bar{\mathbf{C}}_i^* = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}) \cdot \mathbf{X}^T\bar{\mathbf{X}}. \tag{7}$$

4: Calculate the normalized residuals of $\bar{\mathbf{x}}_i \in \bar{\mathbf{X}}$ over all classes by

$$r_j(\bar{\mathbf{x}}_i) = \frac{\|\bar{\mathbf{x}}_i - \mathbf{X}\delta_j(\mathbf{c}_i^*)\|_2}{\|\delta_j(\mathbf{c}_i^*)\|_2}. \tag{8}$$

5: Assign $\bar{\mathbf{x}}_i$ to the class which produces the minimal residual by

$$\text{identity}(\bar{\mathbf{x}}_i) = \underset{j}{\text{argmin}}\{r_j(\bar{\mathbf{x}}_i)\}. \tag{9}$$

Output: The cluster membership of $\mathbf{X}$ and $\bar{\mathbf{X}}$.

Putting everything together, the computational complexity of SSSC is $O(t_1mp^3 + t_2pk^2 + np^2)$ owing to $k, m < p \ll n$, which is obviously less than that of SSC ($O(t_1mn^3 + t_2nk^2)$). On the other hand, the space complexity of SSSC is only $O(p^2)$, comparing with $O(n^2)$ of SSC.

4. Experimental verification and analysis

In this section, several experiments were conducted to show the effectiveness and efficiency of our Scalable Sparse Subspace Clustering (SSSC[1]).

4.1. Data sets

We carried out our experiments using six real-world data sets which cover facial images, handwritten digital data, news corpus, etc. The data sets include one small-sized data sets, three medium-sized data sets, and two large scale data sets. We presented some statistics on these data sets in Table 2 and a brief description as follows:

Extended Yale Database B (ExYaleB) [12] is a facial database which contains 2414 frontal-face images of 38 subjects (about 64 images for each subject). We cropped

[1]We have provided the MATLAB code of SSSC at "http://www.machineilab.org/users/pengxi/"

Table 2. Data sets used in our experiments.

Data sets	# of instances	Dim.	# of classes
ExYaleB	2414	32256	38
RCV	8293	18933	65
MPIE	8916	8200	286
PenDigits	10992	16	10
Covtype	581012	54	7
PokerHand	1000000	10	10

the images from 192×168 to 48×42 and extracted 114 features by using PCA to retain 98% energy of the cropped data; Multiple PIE (MPIE) [14] contains the facial images of 286 individuals captured in four sessions with simultaneous variations in pose, expression and illumination. The size of cropped images is 50×41 (from 100×82), and the experiments were conducted on the 115 features which preserve 98% information of the data; Reuters-21578 (RCV) [1] is a documental corpus, we performed experiments using 785 features that retain 85% information of the original data. Moreover, we tested SSSC using three UCI data sets[2], i.e., PenDigits, Covtype, and PokerHand. PokerHand is an extreme unbalanced data set, of which the maximal class contains 501,209 samples, comparing with 3 samples of the minimal class. We examined the performance of the algorithms using the original data set (PokerHand-1) and a subset (PokerHand-2) with 971,329 data points over the three largest classes.

4.2. Baselines and Evaluation Metrics

Spectral clustering and kernel-based methods are two popular methods to cope with non-linear separable data, and several studies [10] have established the equivalence between them. In our experiments, we compared SSSC with three accelerating spectral clustering algorithms (KASP [30], Nyström approximation based spectral clustering [11], LSC [4]), one kernel-based approach (AKK [6]), and k-means as a baseline. We examined the two variants of Nyström based methods and LSC, denoted as Nyström, Nyström-Orth, LSC_R, and LSC_K. The approximate affinity matrix of Nyström is non-orthogonal, while that of Nyström-Orth has orthogonalized columns. LSC_R randomly selects landmarks from data set, but LSC_K uses the cluster centers of k-means as landmarks. All algorithms were implemented in MATLAB and ran on an Intel Xeon 2.13GHz processor with 16.00 GB RAM. The Matlab code of SSSC and the used data sets could be downloaded from the website 'https://www.dropbox.com/s/e7e9a16nhvpk4o8/SSSC_code%26Data_MM2013.rar'.

In all experiments, we tuned the parameters of all methods to get their best Accuracy. For SSSC, we used Homotopy optimizer to solve ℓ^1-minimization problem. It needs two user-specified parameters, sparsity parameter λ and error tolerance parameter δ. We found a good value in the ranges of $\lambda = (10^{-7}, 10^{-6}, 10^{-5})$ and $\delta = (10^{-3}, 10^{-2}, 10^{-1})$. It should be pointed out that SSSC does not introduce new parameter, the number of required parameters only depends on the adopted optimization algorithm. Refer to the parameter configurations in [4, 6, 30], respectively, we found the optimal parameter for the investigated approaches. KASP and Nyström employ heat kernel to calculate the pairwise similarity, which need specifying the width of heat kernel τ. We specified the range of τ is $[0.1, 1]$ with an interval of 0.1 and $[2, 20]$ with an interval of 1; LSC needs the number of nearest landmarks r for a single point. We searched a good r in the range of $[2, 20]$ with an interval of 1; AKK employs RBF kernel with the parameter σ lying in the range $[0.1, 1]$ with an interval of 0.1.

We measured the quality of the competing algorithm via a Accuracy [33] and Normalized Mutual Information (NMI) [1] between the produced clusters and the ground truth categories. The value of accuracy or NMI is 1 indicates perfect matching with the true subspace distribution whereas 0 indicates perfect mismatch.

4.3. Experimental Results

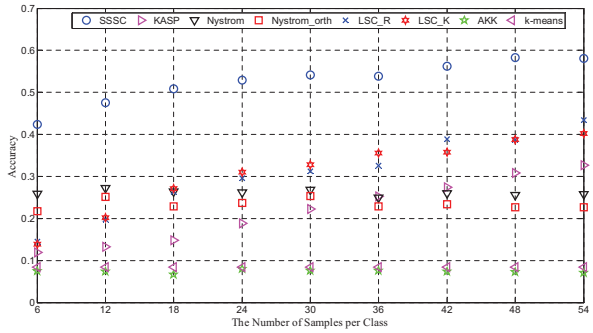
Accuracy and NMI versus varying p: We firstly study the role of in-sample data size in clustering quality. There are two approaches adopted by the investigated algorithms to select some data points as in-sample data, i.e., uniform random sampling (SSSC, Nyström, Nyström_Orth, LSC_R and AKK) and k-means clustering algorithm (KASP and LSC_K). In our experiments, we randomly partitioned each data set into two parts, in-sample data and out-of-sample data. The in-sample data contains $p = 6 \times \tilde{p}$ images from each subject of the ExYaleB database, where $\tilde{p}$ increases from 1 to 9 with an interval of 1. In the same way, we got another nine data sets for KASP and LSC_K by performing k-means to select p in-sample data points. To avoid the difference in data set, we ran different algorithms over the same data partition.

Figure 2(a) and 2(b) report the clustering quality of the proposed method on the ExYaleB with the different in-sample data size. We have the following observations:

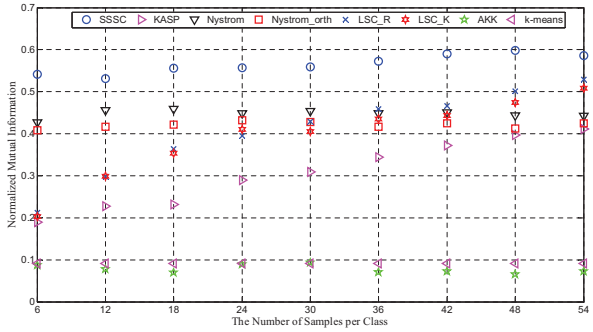
- The accelerating spectral clustering methods (SSSC, Nyström, Nyström_Orth, LSC, and KASP) outperform kernel-based method (AKK) with considerable margin in Accuracy and NMI, and our SSSC achieves the best results in the tests.

- The Accuracy and NMI of AKK are very close to that of k-means. The result is consistent with the report of Chitta et al. [6] that the difference in the clustering errors between standard kernel k-means and AKK will

[2] http://archive.ics.uci.edu/ml/datasets.html



(a) Accuracy versus the different number of in-sample data



(b) NMI versus the different number of in-sample data

Figure 2. Clustering quality of the competing algorithms using the ExYaleB data set. The x-coordinate denotes the number of samples per class.

decrease at the rate of $O(1/p)$, where p is the number of in-sample data.

- All large scale spectral clustering algorithms perform better with the increasing of in-sample data except Nyströms which cope with scalability issues of spectral clustering by reducing the size of affinity matrix but data set. By changing the number of in-sample data, the difference in the max-min Accuracy of SSSC is about 15.91%, compared with 20.80% of KASP, 28.96% of LSC_R, and 26.31% of LSC_K; the differences in max-min NMI are 5.66%, 22.24%, 31.74%, and 30.54%, respectively. The results shows that SSSC is more robust than KASP, LSC_R and LSC_K to the in-sample data size.

Medium-sized data sets: Tables 3-5 report the clustering quality and the time cost of the examined methods. It also lists the tuned parameters when the algorithms achieve these results. We carried out experiments over three medium-sized data sets which contain different data types, i.e., facial image, handwritten digit data and documents. For each data set, we chose 1000 data points as in-sample data. These results reveal a number of interesting points as follows:

Table 3. Performance comparison among different algorithms using **MPIE**. The numbers in the parenthesis are the tuned parameters for achieving the best Accuracy.

Algorithms	Accuracy	NMI	Time(s)
SSSC (1e-6, 0.01)	**61.53%**	**82.81%**	847.0
KASP (0.1)	17.09%	57.54%	1479.83
Nyström (0.7)	47.01%	76.97%	**15.28**
Nyström_Orth (0.7)	51.92%	79.96%	64.80
LSC_R (2)	18.50%	54.90%	62.05
LSC_K (3)	17.50%	56.60%	65.69
AKK (0.1)	11.40%	40.16%	24.60
k-means	14.69%	52.45%	268.54

Table 4. Performance comparison among different algorithms using **PenDigits**.

Algorithms	Accuracy	NMI	Time(s)
SSSC (1e-7, 1e-4)	**81.99%**	**78.37%**	17.02
KASP (4)	77.84%	77.97%	12.48
Nyström (0.4)	77.96%	70.20%	35.93
Nyström_Orth (3)	74.78%	67.59%	6.20
LSC_R (15)	80.09%	76.67%	**5.62**
LSC_K (11)	81.73%	77.34%	7.93
AKK (0.01)	77.02%	69.15%	6.21
k-means	77.05%	69.21%	23.71

Table 5. Performance comparison among different algorithms using **RCV**.

Algorithms	Accuracy	NMI	Time(s)
SSSC (1e-7, 0.01)	**32.40%**	33.81%	1320.63
KASP (0.1)	22.32%	24.79%	198.80
Nyström (0.4)	23.22%	27.55%	27.08
Nyström_Orth (0.1)	25.88%	22.70%	3401.30
LSC_R (2)	14.24%	22.58%	8.87
LSC_K (4)	14.45%	23.69%	17.72
AKK (1)	23.57%	**36.40%**	27.94
k-means	19.05%	26.98%	**0.33**

- Considering the clustering quality, SSSC achieves the best clustering quality over the three data sets except the second best result in NMI over the RCV data set. For example, SSSC achieves a 9.61% gain in Accuracy on MPIE over the second best algorithm. We can see that running times is a main weakness of SSSC when the data set is medium-sized. However, it will be more competitive when we perform SSSC to cluster large scale data set as demonstrated in the next experiment. In Section 3.2, we have shown that the time cost of SSSC also depends on the dimensionality of data set. Therefore, performing SSSC over RCV (785D) is the more time-consuming than over MPIE and PenDigits.

- In most cases, LSC_K outperforms LSC_R with a little improvement, which verifies the claim [17] that complex sampling techniques actually could not produce a better result than random sampling. Consequently,

Table 6. Performance comparison among different algorithms using **Covtype**.

Algorithms	Accuracy	NMI	Time(s)
SSSC (1e-5, 0.1)	**31.05%**	**6.82%**	325.51
KASP (3)	25.36%	3.50%	1314.52
Nyström (0.1)	23.76%	3.79%	**40.61**
Nyström_Orth (0.1)	23.36%	3.98%	351.58
LSC_R (2)	23.24%	6.06%	154.48
LSC_K (4)	25.90%	6.74%	1155.40
AKK (1)	25.36%	3.67%	344.24
k-means	20.84%	3.69%	4895.70

Table 7. Performance comparison among different algorithms using **PokerHand-1** over 10 classes.

Algorithms	Accuracy	NMI	Time(s)
SSSC (1e-5, 0.1)	**19.31%**	0.20%	474.14
KASP (3)	12.37%	0.05%	7049.90
Nyström (0.2)	13.09%	**0.23%**	65.50
Nyström_Orth (18)	16.48%	0.04%	205.70
LSC_R (5)	12.24%	0.00%	1936.80
LSC_K (3)	12.32%	0.00%	8829.00
AKK (0.01)	10.50%	0.03%	2882.50
k-means	10.41%	0.03%	7188.80

Table 8. Performance comparison among different algorithms using **PokerHand-2** over 3 classes.

Algorithms	Accuracy	NMI	Time(s)
SSSC (1e-7,0.2)	**51.60%**	**0.63%**	267.71
KASP (0.3)	35.24%	0.09%	5497.06
Nyström (0.2)	47.90%	0.16%	**61.43**
Nyström_Orth (20)	47.74%	0.00%	204.43
LSC_R (8)	34.91%	0.00%	1891
LSC_K (2)	34.99%	0.00%	8765.5
AKK (0.1)	35.96%	0.62%	1039.28
k-means	36.02%	0.01%	4760.4

with the increasing of data size, a better choice is performing random sampling as preprocessing step for its computational complexity only is $O(1)$.

- In [4], Chen and Cai investigated the Accuracy of LSC_R, LSC_K, Nyström_Orth, and KASP using the PenDigits data set. The best Accuracy of these four algorithms are 79.04%, 79.27%, 73.94% and 72.47%, comparing with 81.73%, 80.09%, 74.78% and 77.84% in our experiments.

Large scale data sets: Tables 6-8 are the results of the competing methods over three large scale data sets. For each data set, we set the size of in-sample data as 1000, and used the remaining data as out-of-sample data. We have the following observations:

- For the large scale data set, NMI failed to measure the performance of the competing algorithms whose NMIs are close to 0.

- SSSC is superior to the other approaches. For example, the Accuracy of SSSC over Covtype is least 5.14% higher than the other tested methods. For PokerHand-1 and PokerHand-2, the gains are 2.83% and 3.71%, respectively.

- In literature, [4] reported that the highest Accuracy over Covtype achieved by LSC_R, LSC_K, Nyström_Orth and KASP are 24.75%, 25.50%, 22.31% and 22.42%, respectively. In our experiments, the Accuracy of the algorithms are 23.24%, 25.90%, 23.36% and 25.36%, respectively. Moreover, KASP performed better in [30] over Pokerhand-2, whose the Accuracy is about 50.11%. The possible reason is that they adopted a more complex random sampling technique.

- With the increasing of data size, the running time of SSSC is no longer a fatal weakness. It benefits from the way to solve scalability issue, i.e., "sampling, clustering, coding and classifying". In the other hand, the used memory of SSSC only depends on the number in-sample data, which makes SSSC feasible to group very large data set, e.g., n is larger than one billion.

5. Conclusion

The representation based spectral clustering algorithms have became more and more popular owing to its effectiveness. However, the over-high computational complexity has hindered its application in practice, especially, the scenario of booming big data. In this paper, we have presented a simple but effective accelerating spectral clustering method, called Scalable Sparse Subspace Clustering (SSSC). SSSC is an out-of-sample extension of Sparse Subspace Clustering (SSC) which makes SSC feasible in large scale setting. Given a data set with n data points, SSSC selects $p \ll n$ data as in-sample data and performs SSC over these data; after that, get the final cluster assignment by coding and classifying non-sampled data based on the sampled data. Extensive experiments show the effectiveness and efficiency of our method comparing the state-of-the-art approaches.

There are some potential ways to improve or extend this work. Although the experimental results show that the proposed algorithm has achieved state-of-the-art results, the recovery conditions of SSSC is largely untouched, especially, when the subspaces are dependent with each other. Moreover, it is also important to develop the error bound of SSSC which is useful to determine the value of in-sample data size.

Acknowledgements: This work was supported by National Basic Research Program of China 973 Program under Grant No.2011CB302201 and National Nature Science Foundation of China under grant No.61003042.

References

[1] D. Cai, X. F. He, and J. W. Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637, 2005.

[2] G. L. Chen and G. Lerman. Spectral curvature clustering (scc). *International Journal of Computer Vision*, 81(3):317–330, 2009.

[3] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):568–586, 2011.

[4] X. Chen and D. Cai. Large scale spectral clustering with landmark-based representation. In *AAAI Conference on Artificial Intelligence*, 2011.

[5] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang. Learning with ℓ^1-graph for image analysis. *IEEE Transactions on Image Processing*, 19(4):858–866, 2010.

[6] R. Chitta, R. Jin, T. Havens, and A. Jain. Approximate kernel k-means: solution to large scale kernel clustering. In *ACM SIGK-DD international conference on Knowledge discovery and data mining*, pages 895–903, 2011.

[7] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797, 2009.

[8] E. Elhamifar and R. Vidal. Sparse manifold clustering and embedding. In *Advances in Neural Information Processing Systems*, pages 55–63, 2011.

[9] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *Arxiv preprint arXiv:1203.1005*, 2012.

[10] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta. A survey of kernel and spectral methods for clustering. *Pattern Recognition*, 41(1):176–190, 2008.

[11] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.

[12] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[13] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. *IEEE Conference on Computer Vision and Pattern Recognition*, 1(7):2032–2037, 2007.

[14] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[15] X. He, D. Cai, S. Yan, and H. Zhang. Neighborhood preserving embedding, 2005.

[16] X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, volume 16, page 153, 2003.

[17] T. O. Kvalseth. Entropy and correlation: Some comments. *IEEE Transactions on Systems, Man and Cybernetics*, 17(3):517–519, 1987.

[18] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2012.

[19] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1546–1562, 2007.

[20] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[21] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14:849–856, 2002.

[22] F. Nie, Z. Zeng, T. I. W., D. Xu, and C. Zhang. Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. *IEEE Transactions on Neural Networks,*, 22(11):1796–1808, 2011.

[23] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–403, 2000.

[24] X. Peng, L. Zhang, and Z. Yi. Constructing l2-graph for subspace learning and segmentation. *CoRR*, abs/1209.0841, 2012.

[25] L. S. Qiao, S. C. Chen, and X. Y. Tan. Sparsity preserving projections with applications to face recognition. *Pattern Recognition*, 43(1):331–341, 2010.

[26] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011.

[27] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.

[28] L. Wang, C. Leckie, R. Kotagiri, and J. Bezdek. Approximate pairwise clustering for large data sets via sampling plus extension. *Pattern Recognition*, 44(2):222–235, 2011.

[29] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[30] D. H. Yan, L. Huang, and M. I. Jordan. Fast approximate spectral clustering. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 907–916, 2009.

[31] A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Fast l1-minimization algorithms and an application in robust face recognition: A review. Technical Report UCB/EECS-2010-13, EECS Department, University of California, Berkeley, February 5 2010.

[32] L. Zhang, M. Yang, and X. Feng. Sparse representation or collaborative representation: Which helps face recognition? In *IEEE International Conference on Computer Vision*, 2011.

[33] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. *Machine Learning*, 2001.